PROCEEDINGS OF SCIENCE

Working with 10 Gigabit Ethernet

Richard Hughes-Jones [1]**,**
The School of Physics and Astronomy,
The University of Manchester,
Manchester,
M13 9PL
UK
E-mail: R.Hughes-Jones@manchester.ac.uk

Stephen Kershaw
The School of Physics and Astronomy,
The University of Manchester,
Manchester,
M13 9PL
UK
E-mail: stephen.kershaw@manchester.ac.uk

Network technology is always moving forward and with the recent availability of 10 Gigabit Ethernet (10GE) hardware we have a standard technology that can compete in terms of speed with core or backbone Internet connections. This technology can help deliver high-speed data to the end-user but will systems that are currently used with Gigabit Ethernet deliver with 10GE?

We investigate the performance of 10 Gigabit Ethernet network interface cards in modern server quality PC systems. We report on the latency, jitter and achievable throughput and comment on the performance of transport protocols at this higher speed.

Lighting the Blue Touchpaper for UK e-Science - Closing Conference of ESLEA Project
The George Hotel, Edinburgh, UK
26-28 March, 2007

[1] Speaker

1. Introduction

Current developments in Radio Astronomy being researched by the EXPReS [1] project will require multi-gigabit flows across Europe using the National Research Networks interconnected by GÉANT. This paper reports on detailed measurements to determine of the performance and behaviour of 10 Gigabit Ethernet NICs when used in server quality PCs.

2. Methodology

The methodology follows that described in [2]; measurements were made using two PCs with the NICs directly connected together with suitable fibre or copper CX4 cables. UDP/IP frames were chosen for the tests as they are processed in a similar manner to TCP/IP frames, but are not subject to the flow control and congestion avoidance algorithms defined in the TCP protocol and thus do not distort the base-level performance. The packet lengths given are those of the user payload[1].

2.1 Latency

To measure the round trip latency, UDPmon [3] was used on one system to send a UDP packet requesting that a response of the required length be sent back by the remote end. Each test involved measuring many (~1M) request-response singletons. The individual request-response times were measured by using the CPU cycle counter on the Pentium [3] and the minimum, average and maximum times were computed. For all the latency measurements the interrupt coalescence of the network interface cards (NICs) was turned off. The measurements thus provide a clear indication of the behaviour of the host, NIC and the network.

Transfer Element	**Inverse data transfer rate µs/byte**	**Expected slope µs/byte**
Memory access	0.00004	
8 lanePCI-Express	0.000054	
10 Gigabit Ethernet	0.0008	
Memory, PCI-Express & 10 Gigabit		0.00268

Figure 1. Table of the slopes expected for PCI-Express and 10 Gigabit Ethernet transfers.

The latency was plotted as a function of the frame size of the response. The slope of this graph is given by the sum of the inverse data transfer rates for each step of the end-to-end path [2]. Figure 1 shows a table giving the slopes expected for PCI-Express and 10 Gigabit Ethernet transfers. The intercept gives the sum of the propagation delays in the hardware components and the end system processing times. Histograms were also made of the singleton request-response measurements. These histograms show any variations in the round-trip latencies, some of which may be caused by other activity in the PCs.

[1] Allowing for 20 bytes of IP and 8 bytes of UDP headers, the maximum user payload for an Ethernet interface with a 1500 byte Maximum Transfer Unit (MTU) would be 1472 bytes.

2.2 UDP Throughput

The UDPmon tool was used to transmit streams of UDP packets at regular, carefully controlled intervals and the throughput and packet dynamics were measured at the receiver. On an unloaded network, UDPmon will estimate the capacity of the link with the smallest bandwidth on the path between the two end systems. On a loaded network, the tool gives an estimate of the available bandwidth. These bandwidths are indicated by the flat portions of the curves.

In these tests, a series of user payloads from 1000 to 8972 bytes were selected and for each packet size, the frame transmit spacing was varied. For each point, the following information was recorded:

- the throughput;
- the time to send and the time to receive the frames;
- the number of packets received, the number lost, and the number out of order;
- the distribution of the lost packets;
- the received inter-packet spacing;
- the CPU load and the number of interrupts for both transmitting and receiving systems.

The “wire”[2] throughput rates include an extra 66 bytes of overhead and were plotted as a function of the frame transmit spacing. On the right hand side of the plots, the curves show a 1/t behaviour, where the delay between sending successive packets is the most important factor. When the frame transmit spacing is such that the data rate would be greater than the available bandwidth, one would expect the curves to be flat (often observed to be the case).

2.3 TCP Throughput

The Web100 [5] patch to the Linux 2.6.20 kernel was used to instrument the TCP stack allowing investigation of the behaviour of the TCP protocol when operating on a 10 Gigabit link. Plots of the throughput, TCP Congestion window (Cwnd), the number of duplicate acknowledgements (DupACK) and the number of packets re-transmitted were made as a function of time though the flow. A further patch to the kernel allowed incoming TCP packets to be deliberately dropped.

3. Hardware

The Supermicro [6] X7DBE motherboard was used for most of the tests. It was configured with two dual-core 2 GHz Xeon 5130 Woodcrest processors, 4 banks of 530 MHz FD memory and has three 8-lane PCI-Express buses connected via the Intel 5000P MCH north bridge, as shown in the left hand block diagram of Figure 2. Each processor is connected by an independent 1.33GHz front side bus. For one of the tests, a Supermicro X6DHE-G2 motherboard was used at one end of the link. This had two 3.2 GHz Xeon CPUs with a shared

[2] The 66 “wire” overhead bytes include: 12 bytes for inter-packet gap, 8 bytes for the preamble and Start Frame Delimiter, 18 bytes for Ethernet frame header and CRC and 28 bytes of IP and UDP headers

800 MHz front side bus to the Intel 7520 chipset. It has two 8-lane PCI-Express buses, as indicated in the right hand block diagram of Figure 2.

Myricom [7] 10 Gigabit Ethernet NICs were used for all of the tests. These are 8-lane PCI-Express devices and both fibre and copper CX4 versions were tested. Version 1.2.0 of the Myricom myri10ge driver and version 1.4.10 of the firmware was used in all the tests. Also check summing was performed on the NIC and Message Signalled Interrupts were in use for all of the tests.

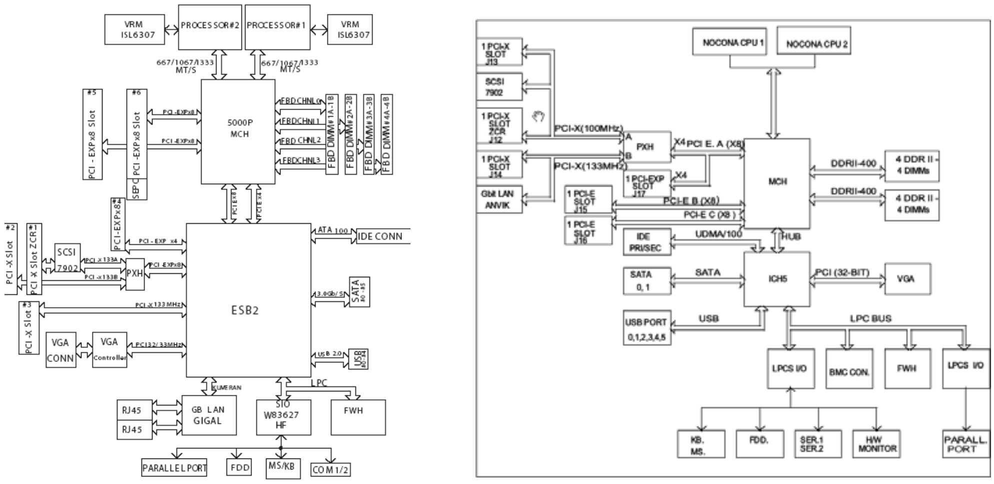

Figure 2. Block diagrams of the Supermicro motherboards used in the tests.
Left: X7DBE dual-core Xeon motherboard Right: X6DHE-G2 motherboard.

4. Measurements made with the Supermicro X7DBE Motherboard

4.1 Latency

Figure 3 shows that variation of round trip latency with the packet size is a smooth linear function as expected, indicating that the driver-NIC buffer management works well. The clear step increase in latency at 9000 bytes is due to the need to send a second partially filled packet. The observed slope of 0.0028 μs/byte is in good agreement with the 0.00268 μs/byte given in Figure 1. The intercept of 21.9 μs is reasonable given the NIC interrupts the CPU for each packet received.

Figure 3 also shows histograms of the round trip times for various packet sizes. There is no variation with packet size, all having a FWHM of ~1 μs and no significant tail.

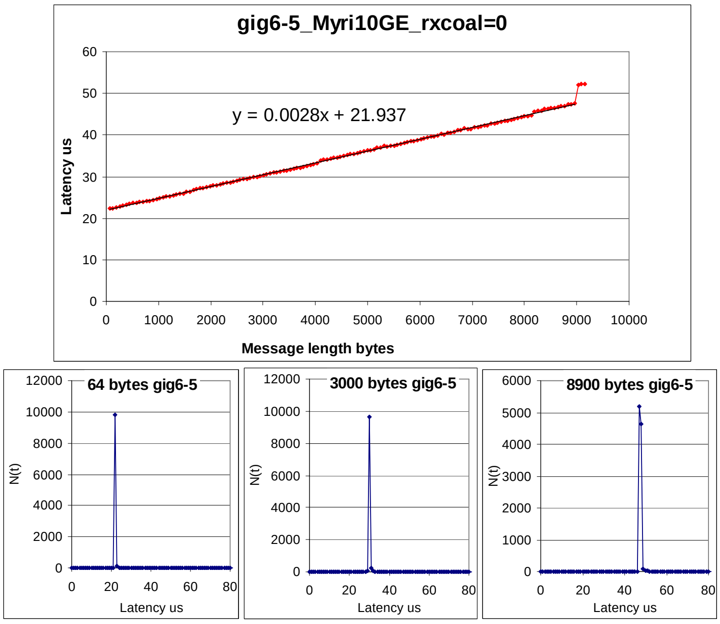

Figure 3. Top: The UDP Request-Response latency as a function of packet size.
Bottom: Histograms of the latency for 64, 300 and 8900 byte packet sizes.

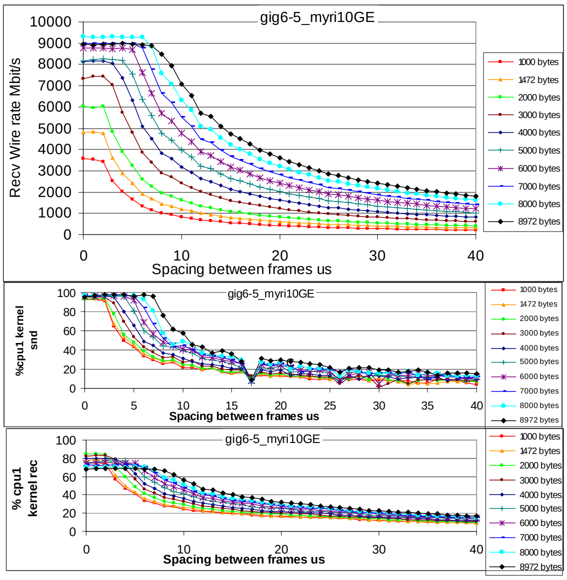

Figure 4. Top: UDP throughput as a function of inter-packet spacing for various packet sizes.
Middle: Percentage of time the sending CPU was in kernel mode.
Bottom: Percentage of time the receiving CPU was in kernel mode.

4.2 UDP Throughput

For these measurements the interrupt coalescence was set to the default value of 25 µs. Figure 4 shows that the NICs and host systems performed very well at multi-gigabit speeds, giving a maximum throughput of 9.3 Gbit/s for back to back 8000 byte packets. For streams of 10 M packets, about 0.002% packet loss was observed in the receiving host. For packets with spacing of less than 8 µs, one of the four CPU cores was in kernel mode over 90% of the time, and the other three CPUs were idle. Similarly for the receiving node, where one of the four CPU cores was in kernel mode 70-80% of the time and the other three CPUs were idle. As the packet size was reduced, processing and PCI-Express transfer overheads become more important and this decreases the achievable data transfer rate.

It was noted that the throughput for 8970 byte packets was less than that of 8000 byte packets, so the UDP achievable throughput was measured as a function of the packet size. The results are shown in Figure 5.

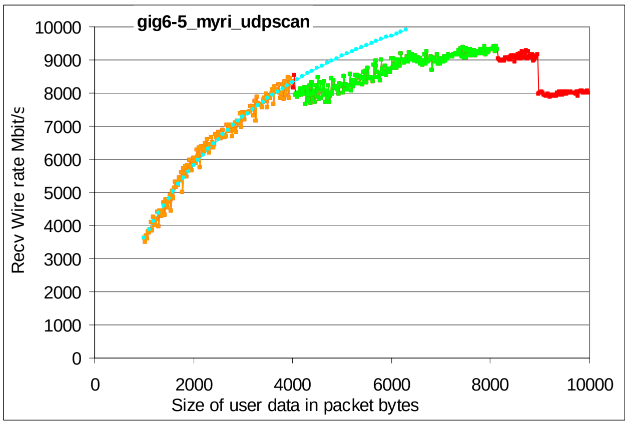

Figure 5. Measurement of UDP throughput as a function of packet size.

5. Measurements made with the SuperMicro X6DHE-G2 Motherboard

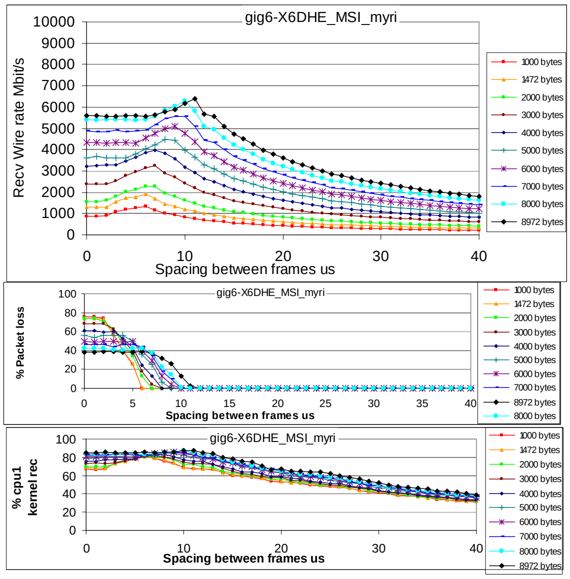

Figure 6. UDP throughput as a function of inter-packet spacing for various packet sizes using the Supermicro XDHE-G2 motherboard.
Middle: Percentage of time the sending CPU was in kernel mode.
Bottom: Percentage of time the receiving CPU was in kernel mode.

Figure 6 shows the achievable UDP throughput and CPU usage when packets are sent from a Myricom NIC in a Supermicro X7DBE motherboard to one in a X6DHE-G2 motherboard. The maximum throughput is only 6.5 Gbit/s with no clear plateau indicating a simple bottleneck. Given the lower CPU usage for the sending CPU than that shown in Figure 4, it is possible that limitations in moving received packets from the NIC to the memory result in queues building up in the receiving NIC, which then sends Ethernet pause packets to the sender. Clearly not all motherboard and chipsets provide the same input-output performance.

6. Protocol Performance

6.1 TCP flows

Plots of the parameters taken from the web100 interface to the TCP stack [2] are shown in Figure 7 for a memory to memory TCP flow generated by iperf and demonstrate the congestion avoidance behaviour in response to lost packets. This TCP flow was set up between two systems connected back-to-back and used a TCP buffer size of 256 kbytes, which is just smaller than the bandwidth-delay product, BDP, of 300 kbytes. The packets were deliberately dropped using a kernel patch in the receiving host. The upper plot in Figure 7 shows Cwnd decreasing by half when a lost packet is detected by the reception of multiple duplicate acknowledgments (DupACKs), shown in the second plot. The third plot shows that one packet is re-transmitted for each packet dropped, while the bottom plot indicates that there is not much reduction in the achievable TCP throughput. This is due to the short round trip time when the systems are connected back-to-back.

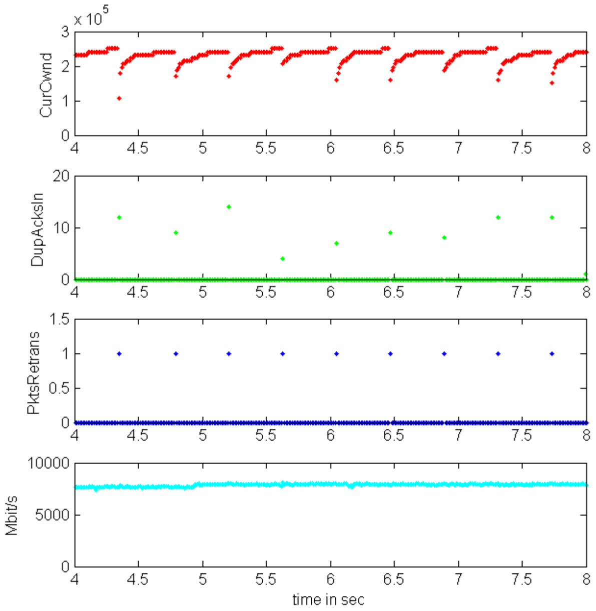

Figure 7. Parameters from the Reno TCP stack recorded by Web100 for an iperf flow. Packets were dropped in the receiving kernel.

6.2 UDP flows with Concurrent Memory Access

As discussed in section 4.2 and shown in Figure 4, a 9.3 Gbit/s UDP flow uses one of the four CPU cores in kernel mode 70-80% of the time, but the other three were unused. Tests were made to determine if the other three CPUs could be used for computation at the same time as sustaining a multi-gigabit flow. Figure 8 shows measurement of the achieved UDP throughput and packet loss under three conditions for a series of trials. On the left are the results for just a UDP flow, in the centre a process that continually accesses memory was run on the second core of the CPU chip processing the networking, and on the right this process was run on the second CPU. In both cases when the load process was run there was a reduction of ~200 Mbit/s in the throughput and about 1.5% packet loss. Figure 9 shows the percentage of time the four CPUs were in different modes when the memory load process was run on CPU3. These results demonstrate that useful work can be done in the end host with minimal effect on the network flow.

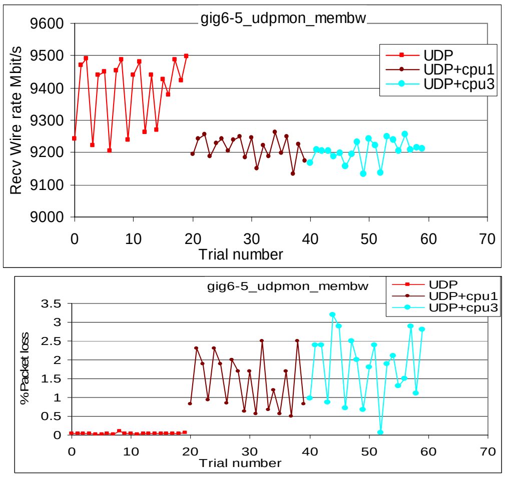

Figure 8. The UDP thoughput and packet loss for a network flow only and when a CPU-Memory process is run on another CPU.

```
Cpu0  :  6.0% us, 74.7% sy,  0.0% ni,   0.3% id,  0.0% wa,  1.3% hi, 17.7% si,  0.0% st
Cpu1  :  0.0% us,  0.0% sy,  0.0% ni, 100.0% id,  0.0% wa,  0.0% hi,  0.0% si,  0.0% st
Cpu2  :  0.0% us,  0.0% sy,  0.0% ni, 100.0% id,  0.0% wa,  0.0% hi,  0.0% si,  0.0% st
Cpu3  : 100.0% us, 0.0% sy,  0.0% ni,   0.0% id,  0.0% wa,  0.0% hi,  0.0% si,  0.0% st
```

Figure 9. The percentage of time the four CPUs were in different modes when the memory load process was run on CPU3.

7. Conclusions

This work has demonstrated that the Myricom 10 Gigabit Ethernet NICs can deliver UDP flows of 9.3 Gbit/s and TCP flows of 7.77 Gbit/s when using PCs using the Supermicro X7DBE. However not all motherboard and chipsets provide the same input-output performance.

Even though one of the CPU cores is occupied in driving the network, these results also show that useful work can be done in the other CPUs. We conclude that these Myricom- X7DBE will be suitable to evaluate the performance of 4 Gigabit UDP flows over lightpaths provisioned over the GÉANT2 network.

References

[1] EXPReS, Express Production Real-time e-VLBI Service Three year project, started March 2006, funded by the European Commission (DG-INFSO), Sixth Framework Programme, Contract #026642 www.express-eu.org .

[2] R. Hughes-Jones, P. Clarke, S. Dallison, "Performance of 1 and 10 Gigabit Ethernet Cards with Server Quality Motherboards," Future Generation Computer Systems Special issue, 2004

[3] UDPmon: a Tool for Investigating Network Performance, http://www.hep.man.ac.uk/~rich/net

[4] R. Hughes-Jones and F. Saka, *Investigation of the Performance of 100Mbit and Gigabit Ethernet Components Using Raw Ethernet Frames*, Technical Report ATL-COM-DAQ-2000-014, Mar 2000. http://www.hep.man.ac.uk/~rich/atlas/atlas_net_note_draft5.pdf

[5] web100 interface to the TCP stack Web100 Project home page, http://www.web100.org/

[6] SuperMicro motherboard reference material, http://www.supermicro.com/products/motherboard/matrix/

[7] Myricom home page http://www.myri.com/